

Brent O. Hatch (5715)
Mark F. James (5295)
HATCH, JAMES & DODGE
10 West Broadway, Suite 400
Salt Lake City, Utah 84101
Telephone: (801) 363-6363
Facsimile: (801) 363-6666

Stuart H. Singer (admitted pro hac vice)
BOIES, SCHILLER & FLEXNER LLP
401 East Las Olas Boulevard – Suite 1200
Ft. Lauderdale, Florida 33301
Telephone: (954) 356-0011
Facsimile: (954) 356-0022

Attorneys for The SCO Group, Inc.

Robert Silver (admitted pro hac vice)
Edward Normand (admitted pro hac vice)
BOIES, SCHILLER & FLEXNER LLP
333 Main Street
Armonk, New York 10504
Telephone: (914) 749-8200
Facsimile: (914) 749-8300

Stephen N. Zack (admitted pro hac vice)
BOIES, SCHILLER & FLEXNER LLP
Bank of America Tower – Suite 2800
100 Southeast Second Street
Miami, Florida 33131
Telephone: (305) 539-8400
Facsimile: (305) 539-1307

**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF UTAH**

THE SCO GROUP, INC.

Plaintiff/Counterclaim-Defendant,

v.

INTERNATIONAL BUSINESS
MACHINES CORPORATION,

Defendant/Counterclaim-Plaintiff.

**SCO'S MEMORANDUM IN SUPPORT
OF ITS MOTION FOR
RECONSIDERATION OF THE
ORDER DENYING SCO'S MOTION
FOR RELIEF FOR IBM'S
SPOILIATION OF EVIDENCE**

**FILED IN REDACTED FORM
[ORIGINALLY FILED UNDER SEAL]**

Case No. 2:03CV0294DAK
Honorable Dale A. Kimball
Magistrate Judge Brooke C. Wells

SEALED

FILED IN UNITED STATES DISTRICT
COURT, DISTRICT OF UTAH

MAR 16 2007

BY D. MARK JONES, CLERK

DEPUTY CLERK

Brent O. Hatch (5715)
Mark F. James (5295)
HATCH, JAMES & DODGE
10 West Broadway, Suite 400
Salt Lake City, Utah 84101
Telephone: (801) 363-6363
Facsimile: (801) 363-6666

Robert Silver (admitted pro hac vice)
Edward Normand (admitted pro hac vice)
BOIES, SCHILLER & FLEXNER LLP
333 Main Street
Armonk, New York 10504
Telephone: (914) 749-8200
Facsimile: (914) 749-8300

Stuart H. Singer (admitted pro hac vice)
BOIES, SCHILLER & FLEXNER LLP
401 East Las Olas Boulevard - Suite 1200
Ft. Lauderdale, Florida 33301
Telephone: (954) 356-0011
Facsimile: (954) 356-0022

Stephen N. Zack (admitted pro hac vice)
BOIES, SCHILLER & FLEXNER LLP
Bank of America Tower - Suite 2800
100 Southeast Second Street
Miami, Florida 33131
Telephone: (305) 539-8400
Facsimile: (305) 539-1307

Attorneys for The SCO Group, Inc.

**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF UTAH**

THE SCO GROUP, INC.

Plaintiff/Counterclaim-Defendant,

v.

**INTERNATIONAL BUSINESS
MACHINES CORPORATION,**

Defendant/Counterclaim-Plaintiff.

**SCO'S MEMORANDUM IN SUPPORT
OF ITS MOTION FOR
RECONSIDERATION OF THE
ORDER DENYING SCO'S MOTION
FOR RELIEF FOR IBM'S
SPOILIATION OF EVIDENCE**

[FILED UNDER SEAL]

Case No. 2:03CV0294DAK
Honorable Dale A. Kimball
Magistrate Judge Brooke C. Wells

TABLE OF CONTENTS

	<u>Page</u>
PRELIMINARY STATEMENT	1
ARGUMENT.....	3
I. RECONSIDERATION IS WARRANTED WHERE THERE IS A MISAPPREHENSION OF FACTS OR CONTROLLING LAW.....	3
II. CMVC DOES NOT CONTAIN THE EVIDENCE THAT SCO CONTENTS IBM DESTROYED.....	3
A. IBM's Representation That Sandboxes Are Only Used by AIX Developers Is False.....	4
B. IBM's Representation That CMVC Contains the Same Evidence As the Sandboxes Is False.....	7
C. SCO's Claims Regarding Evidence Lost in the Sandbox Purge Are Consistent with Statements Made by SCO When it Moved to Compel Production of CMVC.....	11
D. IBM Has Not Followed Through on The Magistrate Court's Request to Identify Where in CMVC the Evidence SCO Contends Was Destroyed Can Be Found.. ..	12
III. SCO HAS SUBMITTED COMPELLING PROOF THAT DESTRUCTION OF RELEVANT EVIDENCE OCCURRED.....	13
IV. SCO WAS PREJUDICED BY IBM'S DESTRUCTION OF EVIDENCE.....	16
V. SCO SHOWED THAT IBM ACTED IN BAD FAITH.....	19
CONCLUSION.....	23

TABLE OF AUTHORITIES

	<u>Page</u>
<u>Cases</u>	
<u>Adams v. Gateway, Inc.</u> , 2006 WL 2563418 (D. Utah 2006)	21
<u>Brown & Williamson Tobacco Corp. v. Jacobson</u> , 644 F. Supp. 1240 (N.D. Ill. 1986)	21
<u>Brown & Williamson Tobacco Corp. v. Jacobson</u> , 827 F.2d 1119 (7th Cir. 1987)	21
<u>Cabinetware, Inc. v. Sullivan</u> , Civ. S. 90-313, LKK 1991 WL 327959 (E.D. Cal. 1991)	18, 19
<u>Computer Associates Int'l, Inc. v. Am. Fundware, Inc.</u> , 133 F.R.D. 166 (D. Colo. 1990)	21
<u>Hancock v. City of Oklahoma City</u> , 857 F.2d 1394 (10th Cir. 1998)	3
<u>In re Napster, Inc. Copyright Litig.</u> , 462 F. Supp. 2d 1060 (N.D. Cal. 2006)	21
<u>Mantle Ranches, Inc., v. U.S. Park Serv.</u> , 950 F. Supp. 299 (D. Colo. 1997)	3
<u>Price v. Philpot</u> , 420 F.3d 1158 (10th Cir. 2005)	3
<u>Servants of the Paraclete v. Does</u> , 204 F.3d 1005 (10th Cir. 2000)	3
<u>United States v. Maxfield</u> , No. 04CR00149, 2007 WL 121128, at *1	3

Plaintiff, The SCO Group, Inc. ("SCO"), by and through undersigned counsel, submits this Memorandum in Support of its Motion for Reconsideration of the Order Denying SCO's Motion for Relief for IBM's Spoliation of Evidence.

PRELIMINARY STATEMENT

The Magistrate Court denied SCO's Motion for Relief for IBM's Spoliation of Evidence "for the reasons set forth by the Court at the hearing held on January 18, 2007." (March 2, 2007 Order.) At the January 18 hearing, the Magistrate Court said:

I am going to find that based upon the evidence before me, as it's put into context, reflects that SCO's motion will be denied. It cannot show that any evidence was lost or destroyed. In fact, I find that it is available and has been available through CMVC. The evidence before me, when seen in context, does not show that IBM acted in bad faith nor does the evidence show that it has been prejudiced because the evidence, as I indicated, has been and is reasonably available. So the motion regarding spoliation and the adverse inference instruction will be denied.

(1/18/07 Hrg. Tr., Ex. 1 at 56:2-11.)

SCO respectfully submits that this decision was based on a misapprehension of fact and law, and should be reconsidered on the following grounds:

First, the primary basis for the ruling – that the evidence SCO contends IBM destroyed "is available and has been available through CMVC" – is not correct. IBM's representations at the January 18 hearing in this regard are false. Neither CMVC (the AIX change control system referenced by the Magistrate Court) nor RCS (the Dynix/ptx change control system) contains the evidence SCO contends IBM destroyed.¹ CMVC and RCS could not possibly show whether that

¹ The Magistrate Court only referenced CMVC, which is an AIX code repository. CMVC does not contain any Dynix/ptx code, so could not possibly have contained duplicates of any Dynix/ptx code that

code had been retained on programmers' systems when they transitioned to work on Linux, or what particular subparts or versions of the operating system had been retained by the programmer. Furthermore, those systems could not possibly contain drafts of Linux code.

Second, SCO submitted substantial proof in the form of IBM executive and programmer emails and deposition testimony that IBM destroyed relevant evidence, and that evidence should be reconsidered now that SCO has shown that the destroyed evidence was not available elsewhere.

Third, the Court's finding that SCO was not prejudiced was based on the mistaken conclusion that the evidence was still available to SCO elsewhere. In reality, the evidence destroyed by IBM was not available to SCO in CMVC or RCS. In addition, the evidence destroyed is directly relevant to SCO's claim that IBM's Linux programmers referred to AIX and Dynix/ptx code while programming for Linux and to SCO's efforts to identify the AIX and Dynix/ptx source code behind IBM's disclosures to Linux, including IBM's numerous disclosures of methods and concepts.

Fourth, the Court's finding that IBM did not act in bad faith, for purposes of the adverse inference instruction requested by SCO, was also based on the mistaken conclusion that the evidence was still available to SCO elsewhere. In fact, SCO has shown that IBM directed that highly relevant and original evidence be destroyed just a month after the instant lawsuit was filed. Bad faith must be inferred from this conduct.

SCO contends was destroyed. Therefore, SCO also addresses herein IBM's anticipated argument that any destroyed Dynix/ptx code would have been included on the Dynix/ptx change control system, RCS, comparable to AIX's CMVC.

ARGUMENT

I. RECONSIDERATION IS WARRANTED WHERE THERE IS A MISAPPREHENSION OF FACTS OR CONTROLLING LAW.

Reconsideration is warranted “where the court has misapprehended the facts, a party’s position, or the controlling law.” Servants of the Paraclete v. Does, 204 F.3d 1005, 1012 (10th Cir. 2000). Neither a change in law nor new evidence is required. See, e.g., Mantle Ranches, Inc., v. U.S. Park Serv., 950 F. Supp. 299, 300, 302 (D. Colo. 1997) (granting in part motion to reconsider, though no change in law or new evidence existed). A district court has the discretion to reconsider any “order short of a final decree.” Price v. Philpot, 420 F.3d 1158, 1167 n.9 (10th Cir. 2005). The consideration of the merits of a motion for reconsideration is squarely within that discretion. See United States v. Maxfield, No. 04CR00149, 2007 WL 121128, at *1 (Ex. A) (D. Utah Jan. 11, 2007) (citing Hancock v. City of Oklahoma City, 857 F.2d 1394, 1395 (10th Cir. 1998)).

IBM previously sought relief through a Motion for Reconsideration before this Court in similar circumstances (in a February 11, 2005 motion), and such relief was granted in part (in an April 20, 2005 order). Accordingly, it is well within the discretion of the Magistrate Court to reconsider the denial of SCO’s Motion for Relief for IBM’s Spoliation of Evidence on the grounds raised herein.

II. CMVC AND RCS DO NOT CONTAIN THE EVIDENCE THAT SCO CONTENDS IBM DESTROYED.

Contrary to IBM’s representations at the January 18 hearing, the evidence that SCO contends IBM destroyed in programmer sandboxes is not available. At the hearing, IBM made the following representations to support its contention that, if IBM Linux Technology Center

("LTC") programmers were in fact instructed to purge their sandboxes or other computer workspaces, no evidence was actually destroyed: (1) sandboxes are only used by AIX developers, not by IBM's Linux developers; (2) the CMVC system contains the same evidence that SCO contends was destroyed in the purging of programmer sandboxes; and (3) SCO had previously said CMVC had this evidence. The Magistrate Court ultimately adopted these representations as a core element of the decision, finding that the evidence SCO contends was destroyed "is available and has been available through CMVC." Yet, each of these IBM representations is false, for the reasons set forth below.

Considering the importance of IBM's representations, moreover, the Magistrate Court directed IBM to assist SCO in identifying in CMVC where the duplicate evidence could be found – that is, the evidence lost in the purging of the programmer sandboxes. IBM has not done this, and SCO contends that it cannot be done: Neither CMVC nor RCS contain the evidence that was destroyed in the purging of the programmer sandboxes. The following evidence thus underscores the reasons IBM did not follow up on the request that the Magistrate Court made.

A. IBM's Representation That Sandboxes Are Only Used by AIX Developers Is False.

Sandbox is a term used to describe private computer workspaces in which programmers can draft and test code on which they are working without impacting the entire system. At the hearing, IBM made the following representation to the Court: "Sandboxes, Your Honor, are a development tool that is used in AIX. They are not used in Linux. They are not used in the Linux Technology Center." (Ex. 1 at 37:7-9.) This IBM argument overlooks the facts that the instruction was given to Linux programmers to purge their sandboxes. This instruction would

have made no sense if, as IBM suggests, Linux programmers did not have or use sandboxes. Moreover, also contrary to IBM's representation, those sandboxes or other similar workspaces would have had to have been used by those programmers to develop their code for Linux.

Indeed, IBM's representation that its Linux programmers did not use sandboxes is contradicted by a public interview given by one of IBM's own programmers. (Ex. 2, <http://kerneltrap.org/comment/reply/80>.) IBM programmer William Lee Irwin III explained that he had been employed by Sequent and IBM since 2000, and that he worked on Dynix/ptx, then AIX, and then Linux. When asked whether he had "any tips for the aspiring kernel hacker" (Linux programmer), Mr. Irwin responded:

A more effective approach appears to be creating a sandbox where the data is disposable and the system nonessential and running the code and figuring out what went wrong when the system crashes.

(Id.) Directly contrary to IBM's representations, this IBM Linux programmer shared his experience on using sandboxes when developing for Linux and advised others to do the same.²

SCO's computer science expert Marc Rochkind has also confirmed that "IBM programmers for Dynix/ptx and Linux, as well as AIX, would have had to use sandboxes, or other similar workspaces, to draft, revise and implement computer code for those systems."

(Declaration of Marc Rochkind dated 3/16/07 ("Rochkind Decl."), ¶ 8.) He explains:

[I]t would be incredibly difficult, if not impossible, to write and revise computer code without using a separate workspace, such as a sandbox, in which to implement and test those revisions. It is essential that all details of a proposed operating feature or patch be worked out in an environment separate from the online or official

² This same IBM programmer was identified in SCO's December 2005 Submission as a significant contributor of misused material to Linux.

copy of the source code. This separate environment, the private programmer workspace, is often called a sandbox.

(Id. ¶ 7.)

SCO's computer science expert Evan Ivie has also confirmed that "IBM programmers for Dynix/ptx and Linux, as well as AIX, used sandboxes, or other similar workspaces or programming environments, to draft, revise and implement computer software for those systems." He explains:

Any creative process requires an environment and facility where that creative process can take place. An artist creates paintings in a studio. A woodcarver creates carvings in a woodshop. Artisans, craftsmen, and skilled workers develop facilities where they can perform their work: workbenches, body shops, bakeries, etc. Programmers are no different.

* * *

If IBM had not adopted and/or developed some type of suitable environment for their programmers, it would have taken them back to the late 1950's and early 1960's and would have made programming an incredibly inefficient and slow process. This would be like taking a tractor away from a farmer and giving him a shovel. Even if IBM had tried it, programmers would have resisted the move.

(Declaration of Dr. Evan Ivie dated 3/16/07, "Ivie Decl.", ¶ 2 and ¶ 5 respectively.)

Accordingly, IBM's argument that sandboxes are not used in Linux makes no sense and cannot not somehow amend its instruction to its Linux programmers to purge their sandboxes. That instruction would have impacted all IBM programmers working in the LTC.

B. IBM's Representation That CMVC Contains the Same Evidence As the Sandboxes Is False.

At the hearing, IBM further argued that the evidence that had been in the purged sandboxes was still available to SCO on the CMVC system that IBM had produced to SCO. IBM made the following representations at the hearing regarding the CMVC system it produced to SCO:

Now [SCO's] premise is, well, we may not have lost source code, but what we lost is the ability no figure out which particular programmers had looked at or had access to which particular AIX or Dynix source code. That, Your Honor, is wrong. SCO has that information in spades. Precisely as you pointed out, that was the purpose of producing CMVC. If SCO really cared, if SCO really wanted to know what code a particular developer had on their machine, it would be a very simple exercise to find out. SCO could have easily taken the list of individuals we provided of people who made contributions to AIX and Dynix and compared it to anyone of the other lists we provided them that identified people who made Linux contributions or who worked in the LTC and could have determined if any names were the same. Having determined that names were the same, SCO could have but chose not to turn the [CMVC] machine on and look at it. Because had they done so, they would have been able to figure out exactly what developer X looked at and when, what was checked out, when it was checked out, and what developer X did to it.

(Ex. 1 at 45:13-46:8.) IBM further represented that destruction of evidence in a sandbox was “a bit like Your Honor having discarded a courtesy copy of a motion you ruled on a year ago knowing that Matt has a copy.” (*Id.* at 35:23-25.)

This is simply not accurate. Fundamentally, code on CMVC or RCS is not a copy of code in programmer sandboxes. The type of evidence found in CMVC or RCS is entirely different from what would be on a programmer sandbox. While IBM has contended that a change management control system, such CMVC or RCS, shows what code had been “checked

out” by programmers, those systems could not possibly show whether that code had been retained on programmers systems when they transitioned to work on Linux, or what particular subparts or versions of the operating system had been retained by the programmer. The specific AIX or Dynix/ptx code that a Linux programmer chose to retain on his system while programming for Linux would have provided important evidence in that SCO could compare that particular code to the programmers’ Linux disclosures. The fact that some code the programmer chose to retain on his system might be duplicated somewhere on a code repository entirely misses the point.

SCO’s computer science experts Marc Rochkind and Evan Ivie have confirmed the impossibility of CMVC or RCS containing the evidence IBM now claims they do. Mr.

Rochkind explains:

[N]either CMVC (as to AIX) nor RCS (as to Dynix/ptx) would show whether Linux programmers had retained AIX and Dynix/ptx on their systems when developing code for Linux, and if so, what parts of the AIX and Dynix/ptx operating systems they retained.

[A] sandbox is the only place where the progression of code drafts can be viewed. For AIX code, CMVC shows the initial code that was checked out, and the final code that was checked back in, but not all the steps in between. The RCS system for Dynix/ptx provides even fewer details than CMVC. Yet it is these steps, these intermediate drafts, – saved only on programmers’ sandboxes – that would have been so important to develop further proof of IBM’s copying.

(Rochkind Decl. ¶¶ 12-13.) Similarly, Dr. Ivie explains:

There are several fundamental flaws in the use [of] a change control system, such as CMVC or RCS, to track a software development effort. A typical change control system allows a programmer to “check out” a module, to modify and test it for

some unspecified amount of time, and then when satisfied to "check it back in" to the system. Perhaps this might be compared to trying to see what is going on in a darkened room with a strobe light. However, the strobe only illuminates a small part of the room (the code checked in and out) and the strobe is controlled by a programmer who may or may not want you to see all that is going on (visibility only at check-in time).

For example, let us assume that the programmer is developing a module for Linux, but is basing it on a module that comes from a contractually-protected operating system owned by another company. If the only visibility that we have is the module after it has been appropriately disguised, then tracing the source becomes much more difficult. Perhaps one could compare this to a body shop for processing stolen cars. It is much easier to prove auto theft if one can find the body shop being used. After a paint job, changes to upholstery, options, accessories, and careful modification of the engine and body numbers, it is much more difficult to identify the theft.

(Ivie Decl. ¶¶ 8-9).

Using another analogy, Mr. Rochkind explains that CMVC is like a library, and the AIX files in CMVC (or Dynix files in RCS) are like books in a library. (Rochkind Decl. ¶ 14.) Just as a library has records of which books have been checked out, when they were checked out, who checked the books out, and when they were checked back in, a change management control system like CMVC or RCS might have records of which files in AIX or Dynix/ptx were checked out, when they were checked out, who checked them out, and when they were checked back in.

(Id. ¶ 14.)

What a library does not record, and could not record, is what was done with the books during the time they were checked out. The records in a library will not show: whether the pages of book were copied while the book was checked out; whether the reader retained pages he copied from the book he checked out long after checking the book back in; whether the reader

later used those pages to write another book; or what the progression of drafts of the readers' other book looked like, such as whether the first drafts were similar to the library book he had copied, and then became progressively dissimilar. (Id. ¶ 15.)

Similarly, Mr. Rochkind explains, none of this information is present in CMVC or RCS. (Id. ¶ 16.) Like a library system, CMVC and RCS do not show what the programmers did with the AIX or Dynix/ptx files on their systems (their "sandboxes") after they checked them out, or even after they checked them back into CMVC or RCS. Specifically, CMVC and RCS do not show: whether the AIX or Dynix/ptx files were copied or saved to the programmers' system after being checked out; whether the programmer retained the copied AIX or Dynix/ptx files on his system long after checking the completed files back into CMVC; whether the programmer ever referred back to those AIX or Dynix/ptx files when he was developing for another operating system, such as Linux; what the progression of drafts of the programmers' code for the other operating system, such as Linux, looked like – whether the first drafts were similar to the AIX or Dynix/ptx files he had copied and retained on his system, and then became progressively dissimilar, until the final version actually bore little resemblance to the AIX or Dynix/ptx files on which it had been based; or whether other programmers made use of the checked-out code by obtaining it in already-checked-out form from the programmer who originally checked it out and saving it to their sandboxes. (Id. ¶ 16).

In contrast, Mr. Rochkind and Dr. Ivie explain, a programmers' sandbox would show all of this information. (Id. ¶ 17; Ivie Decl. ¶ 12.) It would show whether AIX and Dynix/ptx files were present and available to the programmer as he worked on Linux, and if so, which particular code the programmer retained and accessed. It would also show his initial drafts of code and the

progression of drafts after that initial draft, leading to the final version. Plainly, drafts of Linux code would not be available on CMVC or RCS.

In summary, CMVC and RCS simply would not and could not contain the evidence that would have been available on IBM Linux programmer sandboxes or other private computer work spaces.

C. SCO's Claims Regarding Evidence Lost in the Sandbox Purge Are Consistent with Statements Made by SCO When It Moved to Compel Production of CMVC.

At the January 18 hearing, the issue also arose whether SCO, in moving to compel IBM to produce the CMVC system, had previously stated that CMVC contained the same information SCO now contends was destroyed in the purging of programmer sandboxes. Specifically, at the hearing, the Magistrate Court noted that counsel for SCO had in February 2004 represented that the CMVC system would enable SCO to "get every version, every iteration [of AIX]" and that counsel further stated that a confidential IBM document regarding CMVC "gives a simplified description at the bottom saying what it boils down to is that all levels of all files are stored on a central server and are available for updating by those with proper authority." (Ex. 1 at 29:17-20.)³

SCO correctly represented at that February 2004 hearing that CMVC contained information that was important to the development of SCO's case as to IBM's misuse of AIX technology in Linux. SCO did not represent that such information was coterminous with the information SCO now contends IBM destroyed in programmer sandboxes. SCO counsel never contended in 2004 that CMVC or RCS would show which Linux programmers retained AIX or

³ The hearing referenced was a February 6, 2004 hearing. (Ex. 3 at 23:18-24:11.)

Dynix/ptx code on their systems when programming for Linux, or what particular code they retained. SCO counsel also plainly never contended in 2004 that CMVC or RCS would have had drafts of Linux code. Thus, SCO's prior representations regarding CMVC or RCS have no relevance to SCO's contention in the instant motion that the purging of programmer sandboxes destroyed relevant evidence.

In connection with this issue, IBM also repeatedly said at the hearing that SCO must not have even used the CMVC system that was produced—implying that SCO would not know whether it contained the information SCO contends was destroyed in the purging of programmer sandboxes. IBM stated that “it appears that SCO has done absolutely nothing with CMVC,” and “SCO could have but chose not to turn the [CMVC] machine on and look at it.” (Ex. 1 at 34:24-25 and 46:3-4, respectively.) These statements are baseless. SCO did use CMVC extensively. At his deposition, Marc Rochkind clearly testified: “My analysis of AIX, as I say in my report, was based on the CMV – CMVC system that was provided by IBM in the form of an actual physical computer, and all of my access to AIX source during the period when I was preparing my original report came off of that system.” (Ex. 4 at 242:3-8.) Mr. Rochkind further explained in his declaration his extensive use of CMVC to identify material misused by IBM. (Rochkind Decl. ¶ 3.) Nevertheless, as set forth above, neither CMVC nor RCS is a substitute for the evidence that would have been contained in the programmer sandboxes.

D. IBM Has Not Followed Through on the Magistrate Court's Request to Identify Where in CMVC the Evidence SCO Contends Was Destroyed Can Be Found.

Considering the representations that IBM made regarding the availability of evidence on CMVC, the Magistrate Court asked IBM to assist SCO in locating the evidence in CMVC that

SCO contends was destroyed by virtue of the instruction to purge programmer sandboxes.

Specifically, the Magistrate Court directed:

The standard I think is reasonably available. I am going to ask IBM, in the spirit of cooperation, Mr. Shaughnessy, to do what you can or have others do it to see if you can assist in identifying it. That doesn't mean that anything is going to change in terms of the deadlines and the scheduling order cutoffs. But if there is somebody who readily has that information, I would ask you to assist in doing that.

(Ex. 1 at 58:11-18.) Notwithstanding this request by the Magistrate Court, and the clear importance of IBM's representations on this regard to the Magistrate Court's conclusions, IBM has not taken any steps to assist SCO in identifying in CMVC (or RCS) the evidence SCO contends was destroyed. IBM's inaction in this regard speaks volumes about the very representations that were so central to the Magistrate Court's conclusions. Moreover, SCO has submitted the clear evidence discussed above that plainly rebuts IBM's representations. CMVC and RCS – while useful for many purposes – simply do not show the particular AIX or Dynix/ptx files that Linux programmers' systems retained on their systems after they began programming for Linux, and do not contain intermediate drafts of Linux code.

III. SCO HAS SUBMITTED COMPELLING PROOF THAT DESTRUCTION OF RELEVANT EVIDENCE OCCURRED.

SCO submitted substantial proof in the form of IBM executive and programmer emails and deposition testimony that IBM destroyed relevant evidence, and that evidence should be fully considered since SCO has established that the destroyed evidence was not available elsewhere:

Swanberg Email: On April 8, 2003 – one month after SCO’s lawsuit was filed and shortly after the decision to restrict access to AIX and Dynix/ptx source code had been made – Randal Swanberg, a senior IBM executive, sent the following email to IBM managers and team leaders relaying additional instructions:

REDACTED

(Ex. 5.)

Daniel Frye Deposition Testimony: The decision described in the Swanberg email was then carried out more broadly within IBM. Daniel Frye instructed Linux programmers within IBM’s LTC to “purge” or “delete” AIX and Dynix/ptx source code from their local machines and sandboxes. Dan Frye testified:

REDACTED

(Ex. 6 at 92:10-93:1 (emphasis added).) In short, programmers within the LTC, who had previously had access to AIX and Dynix/ptx code, and who had had that access removed, were then further instructed to purge or delete all such code from their machines and to purge and delete the contents of their sandboxes.

Paul McKenney Testimony: Linux programmer (and former Dynix/ptx programmer)

Paul McKenney confirmed that he deleted Dynix/ptx source code from his machine in response to such an instruction. He testified:

REDACTED

⁴ The Read Copy Update (or RCU) technology is one of the major items of Dynix/ptx technology SCO contends was improperly disclosed by IBM.

REDACTED

In short, Mr.

McKenney confirmed that he followed the directive of the OSSC and Dan Frye, and deleted code from his computer following SCO's lawsuit.

IV. SCO WAS PREJUDICED BY IBM'S DESTRUCTION OF EVIDENCE.

The Magistrate Court's finding that SCO was not prejudiced was based on the mistaken conclusion that the evidence was still available to SCO. Because SCO has shown the destroyed evidence is in fact not available on CMVC or RCS, SCO asks that this conclusion be reconsidered. The evidence destroyed by IBM is highly probative to SCO's claims and its destruction creates clear prejudice.

The presence of AIX and Dynix/ptx code in the sandboxes of programmers in IBM's LTC – at their very fingertips – would have further refuted IBM's ongoing assertion that its LTC programmers did not access or rely on Dynix/ptx and AIX source code when they were programming code for Linux. Moreover, the particular subparts and version of the Dynix/ptx or AIX operating systems that the programmer chose to retain would have enabled SCO to provide more specific identification of the AIX or Dynix/ptx code on which the programmers' Linux disclosures was based. Finally, intermediate drafts of Linux code from the sandboxes would have most clearly demonstrated the progression of code from the initial AIX or Dynix/ptx code on the sandbox to the final code that was ultimately disclosed to Linux. Mr. Rochkind has explained this prejudice: "Even by reviewing only the final contributions of code to Linux, I identified a significant instance of AIX code in Linux and multiple instances of Dynix/ptx code

⁵ This is remarkable testimony since Mr. McKenney was, at this time, a Linux programmer and not a Dynix programmer.

in Linux. The availability of drafts from sandboxes would have shown whether even more instances of direct copying existed.” (Rochkind Decl. ¶ 18.) Dr. Ivie concurred. (Ivie Decl. ¶ 15.)

The prejudice to SCO from this destruction is particularly severe in light of the fact that Dynix/ptx methods and concepts improperly disclosed to Linux were stricken from SCO’s case because SCO was not able to identify the Dynix/ptx source code behind the methods and concepts with more specificity. This is because the source code deleted from Linux programmer sandboxes would have substantially helped SCO identify the specific Dynix/ptx source code behind these methods and concepts. By showing what Dynix/ptx code IBM Linux programmers had at their fingertips, the sandboxes or similar workspaces (if they had not been purged) could have substantially narrowed the scope of code that SCO had to review to determine what particular code the programmer had in his mind when he disclosed the method and concept to Linux. This would have made that process of identification of code underlying methods and concepts significantly more practicable. Thus, IBM’s destruction of this sandbox evidence is particularly prejudicial to SCO in light of the June 28, 2006 Order striking Dynix/ptx methods and concepts from SCO’s case (affirmed in November 2006, pending a motion for reconsideration).

An excellent example of this (just one example of many) is the IBM programmer Mr. Irwin. SCO identified Mr. Irwin as making improper disclosures to Linux in Item Numbers 38, 167, 171, 172, 175 of SCO’s December 2005 Submission. These items were all stricken for insufficient specificity. SCO contends that Mr. Irwin’s sandbox would have facilitated the

identification of the Dynix/ptx code behind the methods and concepts he disclosed – the very information SCO was sanctioned for not providing.

SCO has identified numerous other former Dynix/ptx programmers whom IBM transferred to the LTC to program for Linux, and who, SCO contends, improperly contributed methods and concepts to Linux.⁶ Like Mr. Irwin, each of these programmers would likely have used sandboxes, or similar workspaces, in developing Linux. These sandboxes or workspaces would have confirmed (had they not been purged) whether these programmers maintained access to old Dynix/ptx code (as their emails and other comments suggested they did) and which parts of Dynix/ptx they retained. The sandboxes also would have contained the drafts of these programmers' code disclosures to Linux, from which copying could more directly be proven.

Accordingly, the evidence destroyed in the purging of the sandboxes is highly relevant to SCO's claims. The importance of the initial source code from which later drafts were developed, and the prejudice flowing from its destruction, was recognized in a similar case involving copyright infringement. In Cabinetware, Inc. v. Sullivan, No. Civ. S. 90-313 LKK 1991, WL 327959 (E.D. Cal. 1991) (Ex. B), the defendant destroyed source code after receiving a document request for such code. The magistrate judge found that "computer programs can be easily modified to disguise the copying of source codes and that a comparison of [the defendant's] initial source code with [the plaintiff's] source code is of critical importance in this case." Id. at *2. Therefore, the magistrate judge recommended that an adverse instruction be given. The district court agreed that the evidence that was destroyed was "essential to plaintiff's

⁶ The group includes: Dipankar Sarma, Paul McKenney, Jack Vogel, Swaminathan Sivasubramanian, Michael Hohnbaum, Gerrit Huizenga, Patricia Gaughen, Martin Bligh, Suparna Bhattacharya, Badari Pulavarty, Hanna Linder, Tim Wright, James Cleverdon, and Dave Hansen.

case” of copyright infringement, and entered a default judgment on the issue of liability against the defendant. *Id.* at *2, 4. As in the Cabinetware case, the code in IBM Linux programmers’ sandboxes was important to SCO’s case because computer programs can so “easily be modified to disguise the copying of source codes.”

V. SCO SHOWED THAT IBM ACTED IN BAD FAITH.

SCO also seeks reconsideration of the decision that SCO did not show that IBM acted in bad faith. Bad faith can and should be inferred from the circumstances surrounding IBM’s directive to purge programmer sandboxes or workspaces. As IBM became involved in Linux, IBM repeatedly and publicly boasted that its experience in and disclosures from AIX and Dynix/ptx was the critical difference in evolving Linux from a hobbyist system to a commercially-hardening operating system. For example, in an interview with Linux Magazine about the state of the Linux kernel in 2001, IBM programmer Patricia Gaughen stated that Linux

REDACTED

another example, Dan Frye, the Director of the LTC, confirmed in an interview with the Consulting Times that the LTC “wanted skills from across IBM, and we have people from AIX, and OS2 . . . and PTX, and Research and so on.” (Consulting Times, Inside IBM – Dan Frye and the Linux Technology Center, Ex. 9.) Frye also discussed the porting of IBM’s proprietary technology to Linux, stating “[IBM] just add[s] arms and legs and skills to make [projects within Linux] go faster.” (*Id.*)

When SCO realized that IBM was improperly using these Unix-derived works in Linux, SCO confronted IBM regarding the problem without success, and then filed the instant lawsuit

against IBM on March 6, 2003. SCO's initial complaint against IBM made clear that the conduct of the LTC in making disclosures to Linux development was at the heart of the lawsuit.

IBM's destruction of evidence then ensued as a direct response to the lawsuit. After SCO filed its lawsuit, IBM immediately determined that access by its Linux programmers to AIX and Dynix/ptx code should be removed. This shows that there was no confusion on IBM's part as to the significance of the AIX and Dynix/ptx code to which its Linux programmers had access and the nature of the problem alleged by SCO in its complaint. However, IBM did not stop there. IBM's OSSC then decided that Linux programmers whose access had been removed should also "purge" their sandboxes or similar workspaces. That OSSC decision was then implemented in IBM's Linux Technology Center – the very organization within IBM that was tasked with developing the code for Linux that was at issue in SCO's lawsuit.

This rapid succession of events from SCO's lawsuit to the destruction of the sandbox evidence necessarily creates an inference of bad faith. IBM cannot reasonably contend that its destruction of evidence was accidental or coincidental. Rather, the destruction was an intentional act taken in response to the filing of SCO's lawsuit. This establishes willfulness and bad faith. Furthermore, IBM spent the next two years after the lawsuit resisting efforts of SCO to obtain access to the code repository systems CMVC and RCS that it now claims absolve it of any culpability for destroying evidence in 2003.

At the January hearing IBM argued that SCO has not satisfied the bad faith element because "SCO has not come forward with a shred of evidence that any member of the LTC, or Mr. Swanberg, ever entertained the thought that what they were doing by making this very simple request was destroying evidence." But this argument goes far beyond what the law

requires of SCO to show bad faith. Bad faith can be clearly inferred from circumstantial evidence. Adams v. Gateway, Inc., No. 2:02-CV-106 TS, 2006 WL 2563418, at *3 (D. Utah 2006) (Ex. C). In particular, courts have found the timing of the destruction of evidence – whether evidence is destroyed after notice that it could be relevant in litigation – probative to bad faith. See, e.g., In re Napster, Inc. Copyright Litig., 462 F. Supp. 2d 1060, 1070, 1072-74 (N.D. Cal. 2006) (finding duty to maintain and not destroy relevant email and other documents after notice of litigation, and that destruction after notice of litigation is relevant to determination of willfulness or bad faith); Computer Associates Int'l, Inc. v. Am. Fundware, Inc., 133 F.R.D. 166, 169-70 (D. Colo. 1990) (finding deletion of source code by computer program developer in copyright action merited spoliation sanction of default because timing of destruction after notice of litigation); Brown & Williamson Tobacco Corp. v. Jacobson, 644 F. Supp. 1240, 1248-49 (N.D. Ill. 1986) (noting under a totality of the circumstances analysis that selective destruction after notice that a lawsuit is pending may serve as basis for finding bad faith spoliation); see also Brown & Williamson Tobacco Corp. v. Jacobson, 827 F.2d 1119, 1134-35 (7th Cir. 1987) (affirming jury finding in trial court on same).

The decision in Computer Associates International, Inc. v. American Fundware, Inc., 133 F.R.D. 166, 169-70 (D. Colo. 1990), is particularly instructive. In that case, Computer Associates (“CA”) alleged that the defendant, American Fundware (“AF”), improperly copied its software in violation of their agreement. As in the SCO case, CA brought these issues to AF’s attention and, when a resolution could not be reached, filed suit against AF. Id. at 168. Before that time, AF had been destroying all previous versions of the software at issue, other than the current version, and AF continued this practice after the litigation had commenced. Id. CA

moved for sanctions based on this spoliation of evidence. The court imposed a default judgment on AF for its destruction – even though the court found that the practice of deleting old versions “is commonly followed in the industry, for legitimate reasons, and is not inherently wrongful.”

Id. The Court persuasively explained: “It is inconceivable that after the October 1986 [pre-litigation] meeting, AF did not realize that the software in its possession would be sought through discovery. Certainly commencement of the action settles any doubts.” Id. at 169 (emphasis added). Therefore, the court concluded that the destruction was willful and in bad faith, and that the destruction of source code, the “best evidence” of copying, inflicted “the ultimate prejudice.” Id. at 170. In this case – where IBM actually undertook the destruction of evidence in response to the litigation and with clear notice that the evidence would be relevant to SCO’s claims – bad faith should be inferred.

CONCLUSION

Based on the foregoing, SCO respectfully requests that the Magistrate Court reconsider the March 2, 2007 Order, and grant the relief requested by SCO.

DATED this 16th day of March, 2007.

HATCH, JAMES & DODGE, P.C.

Brent O. Hatch

Mark F. James

BOIES, SCHILLER & FLEXNER LLP

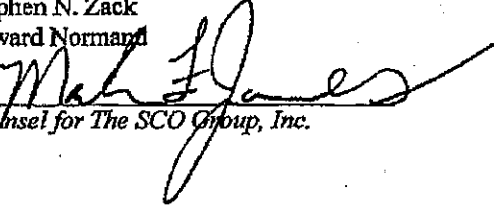
Robert Silver

Stuart H. Singer

Stephen N. Zack

Edward Normand

By


Counsel for The SCO Group, Inc.

CERTIFICATE OF SERVICE

Plaintiff/Counterclaim-Defendant, The SCO Group, Inc., hereby certifies that a true and correct copy of the foregoing **REDACTED SCO'S MEMORANDUM IN SUPPORT OF ITS MOTION FOR RECONSIDERATION OF THE MARCH 2, 2007 ORDER DENYING SCO'S MOTION FOR RELIEF FOR IBM'S SPOILIATION OF EVIDENCE** was served on Defendant/Counterclaim-Plaintiff, International Business Machines Corporation, on this 21st day of March, 2007, via CM/ECF to the following:

David Marriott, Esq. (dmarrriott@cravath.com)
Cravath, Swaine & Moore LLP
Worldwide Plaza
825 Eighth Avenue
New York, New York 10019

Todd Shaughnessy, Esq. (tshaughnessy@swlaw.com)
Snell & Wilmer LLP
1200 Gateway Tower West
15 West South Temple
Salt Lake City, Utah 84101-1004

By  _____